# Intel® Pentium® 4 Processor on 90 nm Process

## Specification Update

*June 22, 2004*

**Notice:** The Intel® Pentium® processor may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are documented in this Specification Update.

Document Number: 302352-003

# Contents



§

# Revision History

| Revision Number | Description | Date |
|---|---|---|
| -001 | • Initial Release | June 2004 |
| -002 | • Added content for Intel® Pentium® 4 processor on 90 nm process in 775-land package<br>• Added 775-land package processor upside marking diagram in Figure 2<br>• Added processor identification information for 775-land package to Table 1<br>• Notes added to clarify that C0 errata only apply to 478 pin package<br>• Modified for Processor Identification information Table Notes | "Out-of-Cycle"<br>June 21 2004 |
| -003 | • Repaired drawings in Figures 1 and 2; reformatted document layout | "Out-of-Cycle"<br>June 22, 2004 |

§

# Preface

This document is an update to the specifications contained in the documents listed in the following Affected Documents/Related Documents table. It is a compilation of device and document errata and specification clarifications and changes, and is intended for hardware system manufacturers and for software developers of applications, operating system, and tools.

Information types defined in the Nomenclature section of this document are consolidated into this update document and are no longer published in other documents. This document may also contain information that has not been previously published.

It is intended for hardware system manufacturers and software developers of applications, operating systems, or tools. It contains S-Specs, Errata, Documentation Changes, Specification Clarifications and Specification Changes.

## Affected Documents

| Document Title | Document Number |
|---|---|
| *Intel® Pentium® 4 Processor on 90 nm Process* datasheet | 300561-001<br>http://developer.intel.com/design/pentium4/datashts/300561.htm |

## Related Documents

| Document Title | Document Number |
|---|---|
| *IA-32 Intel® Architecture Software Developer's Manual Volume 1: Basic Architecture* | http://developer.intel.com/design/pentium4/manuals/253665.htm |
| *IA-32 Intel® Architecture Software Developer's Manual Volume 2A: Instruction Set Reference Manual A–M* | http://developer.intel.com/design/pentium4/manuals/253666.htm |
| *IA-32 Intel® Architecture Software Developer's Manual Volume 2B: Instruction Set Reference Manual, N–Z* | http://developer.intel.com/design/pentium4/manuals/253667.htm |
| *IA-32 Intel® Architecture Software Developer's Manual Volume 3: System Programming Guide* | http://developer.intel.com/design/pentium4/manuals/253668.htm |
| *Intel® Extended Memory 64 Technology Software Developer's Guide Vol 1* | http://developer.intel.com/technology/64bitextensions/300834.htm |
| *Intel® Extended Memory 64 Technology Software Developer's Guide Vol 2* | http://developer.intel.com/technology/64bitextensions/300835.htm |

# Nomenclature

**S-Spec Number** is a five-digit code used to identify products. Products are differentiated by their unique characteristics, e.g., core speed, L2 cache size, package type, etc. as described in the processor identification information table. Care should be taken to read all notes associated with each S-Spec number

**Errata** are design defects or errors. Errata may cause the Intel® Pentium® processor's behavior to deviate from published specifications. Hardware and software designed to be used with any given stepping must assume that all errata documented for that stepping are present on all devices.

**Specification Changes** are modifications to the current published specifications. These changes will be incorporated in the next release of the specifications.

**Specification Clarifications** describe a specification in greater detail or further highlight a specification's impact to a complex design situation. These clarifications will be incorporated in the next release of the specifications.

**Documentation Changes** include typos, errors, or omissions from the current published specifications. These changes will be incorporated in the next release of the specifications.

# General Information

### Figure 1. Intel® Pentium® 4 Processor on 90 nm Process in the 478-pin Package

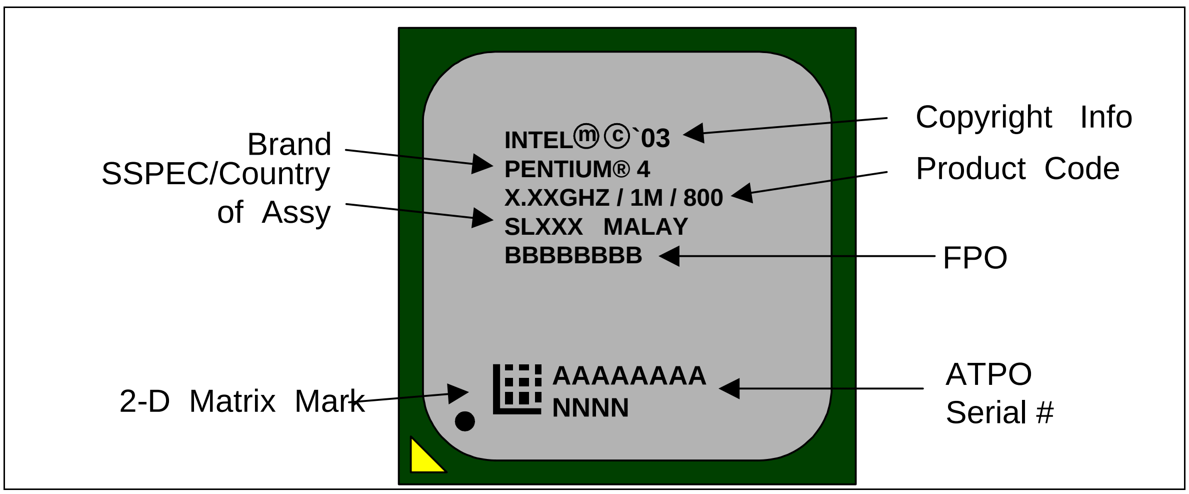
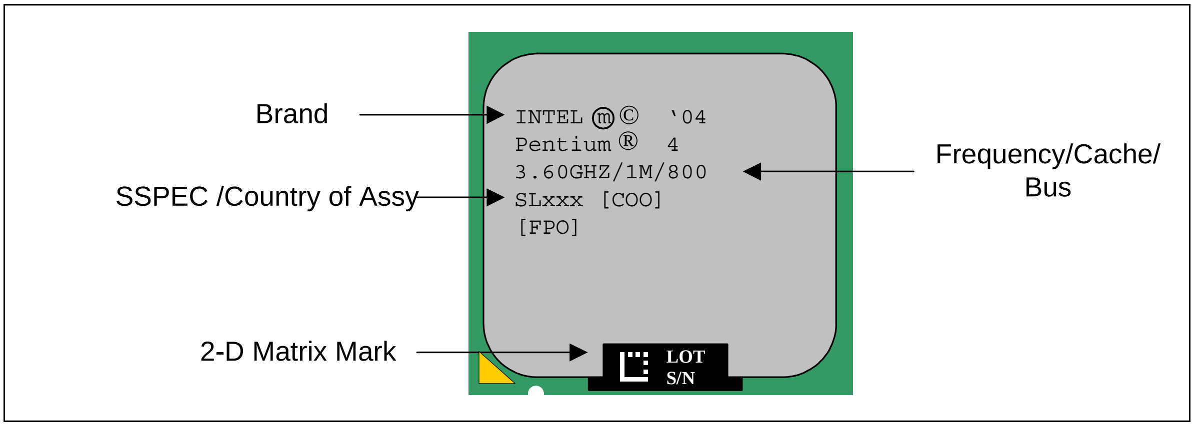


**Figure 2. Intel® Pentium® 4 Processor on 90 nm Process in the 775-Land LGA Package**



# Identification Information

The Pentium 4 processor on 90 nm process can be identified by the following values:

| Family[1] | Model[2] |
|---|---|
| 1111b | 0011b |

**NOTES:**
1. The Family corresponds to bits [11:8] of the EDX register after RESET, bits [11:8] of the EAX register after the CPUID instruction is executed with a 1 in the EAX register, and the generation field of the Device ID register accessible through Boundary Scan.
2. The Model corresponds to bits [7:4] of the EDX register after RESET, bits [7:4] of the EAX register after the CPUID instruction is executed with a 1 in the EAX register, and the model field of the Device ID register accessible through Boundary Scan.

## Table 1. Intel® Pentium® 4 Processor on 90 nm Process Identification Information

| S-Spec | Core Stepping | L2 Cache Size (bytes) | CPUID | Speed Core/Bus | Package and Revision | Notes |
|---|---|---|---|---|---|---|
| SL7B8 | C0 | 1M | 0F33 | 3.20 GHz/800MHz | 35.0 x 35.0 mm FC-mPGA4, Rev 2.0 | 2, 3, 4, 5 |
| SL79L | C0 | 1M | 0F33 | 3.0 GHz/800MHz | 35.0 x 35.0 mm FC-mPGA4, Rev 2.0 | 2, 3, 5 |
| SL79K | C0 | 1M | 0F33 | 2.80 GHz/800MHz | 35.0 x 35.0 mm FC-mPGA4, Rev 2.0 | 2, 3, 5 |
| SL7D8 | C0 | 1M | 0F33 | 2.80 GHz/533MHz | 35.0 x 35.0 mm FC-mPGA4, Rev 2.0 | 3, 6 |
| SL7E8 | C0 | 1M | 0F33 | 2.40 GHz/533MHz | 35.0 x 35.0 mm FC-mPGA4 Rev 2.0 | 1, 5 |
| SL7E6 | D0 | 1M | 0F34h | 3.40GHz/800MHz | 35.0 x 35.0 mm FC-mPGA4 Rev 2.0 | 2, 3, 4 |
| SL7E5 | D0 | 1M | 0F34h | 3.20GHz/800MHz | 35.0 x 35.0 mm FC-mPGA4 Rev 2.0 | 2, 3, 5 |
| SL7E4 | D0 | 1M | 0F34h | 3 GHz/800MHz | 35.0 x 35.0 mm FC-mPGA4 Rev 2.0 | 2, 3, 5 |
| SL7E3 | D0 | 1M | 0F34h | 2.80GHz/800MHz | 35.0 x 35.0 mm FC-mPGA4 Rev 2.0 | 2, 3, 5 |
| SL7E2 | D0 | 1M | 0F34h | 2.80GHz/533MHz | 35.0 x 35.0 mm FC-mPGA4 Rev 2.0 | 3, 6 |
| SL7J9 | D0 | 1M | 0F34h | 3.60GHz/800MHz | 775-land FC-LGA4 37.5 x 37.5 mm Rev 01 | 2, 3, 8 |
| SL7J8 | D0 | 1M | 0F34h | 3.40GHz/800MHz | 775-land FC-LGA4 37.5 x 37.5 mm Rev 01 | 2, 3, 8 |
| SL7J7 | D0 | 1M | 0F34h | 3.20GHz/800MHz | 775-land FC-LGA4 37.5 x 37.5 mm Rev 01 | 2, 3, 7 |

## Table 1. Intel® Pentium® 4 Processor on 90 nm Process Identification Information

| S-Spec | Core Stepping | L2 Cache Size (bytes) | CPUID | Speed Core/Bus | Package and Revision | Notes |
|---|---|---|---|---|---|---|
| SL7J6 | D0 | 1M | 0F34h | 3.0 GHz/800MHz | 775-land FC-LGA4 37.5 x 37.5 mm Rev 01 | 2, 3, 7 |
| SL7J5 | D0 | 1M | 0F34h | 2.80GHz/800MHz | 775-land FC-LGA4 37.5 x 37.5 mm Rev 01 | 2, 3, 7 |
| SL7J4 | D0 | 1M | 0F34h | 2.80GHz/533MHz | 775-land FC-LGA4 37.5 x 37.5 mm Rev 01 | 3, 7 |
| SL7KH | D0 | 1M | 0F34h | 2.80GHz/533MHz | 775-land FC-LGA4 37.5 x 37.5 mm Rev 01 | 1, 3, 7 |
| SL7KJ | D0 | 1M | 0F34h | 2.80GHz/800MHz | 775-land FC-LGA4 37.5 x 37.5 mm Rev 01 | 1, 2, 3, 7 |
| SL7KK | D0 | 1M | 0F34h | 3.0GHz/800MHz | 775-land FC-LGA4 37.5 x 37.5 mm Rev 01 | 1, 2, 3, 7 |
| SL7KL | D0 | 1M | 0F34h | 3.20GHz/800MHz | 775-land FC-LGA4 37.5 x 37.5 mm Rev 01 | 1, 2, 3, 7 |
| SL7KM | D0 | 1M | 0F34h | 3.40GHz/800MHz | 775-land FC-LGA4 37.5 x 37.5 mm Rev 01 | 1, 2, 3, 8 |
| SL7KN | D0 | 1M | 0F34h | 3.60GHz/800MHz | 775-land FC-LGA4 37.5 x 37.5 mm Rev 01 | 1, 2, 3, 8 |

**NOTES:**
1. This is a boxed Pentium 4 processor on 90 nm process with an unattached fan heatsink.
2. These parts include Hyper-Threading Technology.
3. These parts have multiple VIDs.
4. These Pentium 4 processors on 90 nm process support loadline A
5. These Pentium 4 processors on 90 nm process support loadline B
6. These Pentium 4 processors on 90 nm process support loadline B and Hyper-Threading is turned off
7. These Pentium 4 processors on 90 nm process in 775-land LGA package support the 775_VR_CONFIG_04A (mainstream) specifications.
8. These Pentium 4 processors on 90 nm process in 775-land LGA package support the 775_VR_CONFIG_04B (performance) specifications.

§

# Summary Table of Changes

The following table indicates the Errata, Documentation Changes, Specification Clarifications, or Specification Changes that apply to Pentium 4 processors on 90 nm process. Intel intends to fix some of the errata in a future stepping of the component, and to account for the other outstanding issues through documentation or specification changes as noted. This table uses the following notations:

## Codes Used in Summary Table

### Stepping

| | |
|---|---|
| X: | Erratum, Specification Change or Clarification that applies to this stepping. |
| (No mark) or (Blank Box): | This erratum is fixed in listed stepping or specification change does not apply to listed stepping. |

### Status

| | |
|---|---|
| Doc: | Document change or update that will be implemented. |
| Plan Fix: | This erratum may be fixed in a future stepping of the product. |
| Fixed: | This erratum has been previously fixed. |
| No Fix: | There are no plans to fix this erratum. |
| PKG: | This column refers to errata on the Intel® Pentium® 4 processor on 90 nm process substrate. |
| AP: | APIC related erratum. |
| Shaded: | This item is either new or modified from the previous version of the document. |

***Note:*** Each Specification Update item is prefixed with a capital letter to distinguish the product. The key below details the letters that are used in Intel's microprocessor Specification Updates:

A = Intel® Pentium® II processor
B = Mobile Intel® Pentium® II processor
C = Intel® Celeron® processor
D = Intel® Pentium® II Xeon™ processor
E = Intel® Pentium® III processor
G = Intel® Pentium® III Xeon™ processor
H = Mobile Intel® Celeron® processor at 466 MHz, 433 MHz, 400 MHz, 366 MHz, 333 MHz, 300 MHz, and 266 MHz
K = Mobile Intel® Pentium® III Processor – M
M = Mobile Intel® Celeron® processor
N = Intel® Pentium® 4 processor
O = Intel® Xeon™ processor MP
P = Intel® Xeon™ processor
Q = Mobile Intel® Pentium® 4 Processor supporting Hyper-Threading Technology on 90-nm process technology
R = Intel® Pentium® 4 processor on 90 nm process
T = Mobile Intel® Pentium® 4 processor – M
V = Intel® Celeron® processor in the 478-Pin Package

W = Low Voltage Intel® Xeon™ processor
X = Intel® Pentium® M Processor on 90nm process with 2-MB L2 Cache
Y = Intel® Pentium® M Processor
Z = Mobile Intel® Pentium® 4 Processor with 533 MHz System Bus

| NO. | CO[1] | DO | Plan | ERRATA |
|---|---|---|---|---|
| R1 | X | X | No Fix | Transaction Is Not Retried after BINIT# |
| R2 | X | X | No Fix | Invalid Opcode 0FFFh Requires a ModRM Byte |
| R3 | X | X | No Fix | Processor May Hang Due to Speculative Page Walks to Non-Existent System Memory |
| R4 | X | X | No Fix | Memory Type of the Load Lock Different from Its Corresponding Store Unlock |
| R5 | X | X | No Fix | Machine Check Architecture Error Reporting and Recovery May Not Work As Expected |
| R6 | X | X | No Fix | Debug Mechanisms May Not Function as Expected |
| R7 | X | X | No Fix | Cascading of Performance Counters Does Not Work Correctly When Forced Overflow Is Enabled |
| R8 | X | X | No Fix | EMON Event Counting of x87 Loads May Not Work As Expected |
| R9 | X | X | No Fix | System Bus Interrupt Messages without Data Which Receive a HardFailure Response May Hang the Processor |
| R10 | X | X | No Fix | The Processor Signals Page-Fault Exception (#PF) Instead of Alignment Check Exception (#AC) on an Unlocked CMPXCHG8B Instruction |
| R11 | X | X | No Fix | FSW May Not Be Completely Restored after Page Fault on FRSTOR or FLDENV Instructions |
| R12 | X | X | No Fix | Processor Issues Inconsistent Transaction Size Attributes for Locked Operation |
| R13 | X | X | No Fix | When the Processor Is in the System Management Mode (SMM), Debug Registers May Be Fully Writeable |
| R14 | X | X | No Fix | Shutdown and IERR# May Result Due to a Machine Check Exception on a Hyper-Threading Technology Enabled Processor |
| R15 | X | X | No Fix | Processor May Hang under Certain Frequencies and 12.5% STPCLK# Duty Cycle |
| R16 | X | X | No Fix | System May Hang if a Fatal Cache Error Causes Bus Write Line (BWL) Transaction to Occur to the Same Cache Line Address as an Outstanding Bus Read Line (BRL) or Bus Read-Invalidate Line (BRIL} |
| R17 | X | X | No Fix | A Write to APIC Registers Sometimes May Appear to Have Not Occurred |
| R18 | X | | Fixed | Some Front Side Bus I/O Specifications are not Met |
| R19 | X | X | No Fix | Parity Error in the L1 Cache May Cause the Processor to Hang |
| R20 | X | | Fixed | BPM4# Signal Not Being Asserted According to Specification |
| R21 | X | X | No Fix | Sequence of Locked Operations Can Cause Two Threads to Receive Stale Data and Cause Application Hang |
| R22 | X | X | Plan Fix | A 16-bit Address Wrap Resulting from a Near Branch (Jump or Call) May Cause an Incorrect Address to be Reported to the #GP Exception Handler |
| R23 | X | X | No Fix | Bus Locks and SMC Detection May Cause the Processor to Temporarily Hang |
| R24 | X | | Fixed | PWRGOOD and TAP Signals Maximum Input Hysteresis Higher Than Specified |
| R25 | X | X | Plan Fix | Incorrect Physical Address Size Returned by CPUID Instruction |

| NO. | CO[1] | DO | Plan | ERRATA |
|---|---|---|---|---|
| R26 | X | X | No Fix | Incorrect Debug Exception (#DB) May Occur When a Data Breakpoint is set on an FP Instruction |
| R27 | X | X | No Fix | xAPIC May Not Report Some Illegal Vector Errors |
| R28 | X | X | Plan Fix | Enabling No-Eviction Mode (NEM) May Prevent the Operation of the Second Logical Processor in a Hyper-Threading Technology Enabled Processor |
| R29 | X | X | Plan Fix | Incorrect Duty Cycle is Chosen when On-Demand Clock Modulation is Enabled in a Processor Supporting Hyper-Threading Technology |
| R30 | X | X | No Fix | Memory Aliasing of Pages as Uncacheable Memory Type and Write Back (WB) May Hang the System |
| R31 | X | X | Plan Fix | Interactions Between the Instruction Translation Lookaside Buffer (ITLB) and the Instruction Streaming Buffer May Cause Unpredictable Software Behavior |

**NOTES:**
1. Only applies to Pentium 4 processor on 90 nm Process in the 478-pin package

| NO. | C0 | D0 | Plans | SPECIFICATION CHANGES |
|---|---|---|---|---|
| | | | | There are no specification changes s in this Specification Update revision |

| NO. | C0 | D0 | Plans | SPECIFICATION CLARIFICATIONS |
|---|---|---|---|---|
| | | | | There are no specification clarifications s in this Specification Update revision |

| NO. | C0 | D0 | Plans | DOCUMENTATION CHANGES |
|---|---|---|---|---|
| | | | | There are no documentation changes in this Specification Update revision |

# Errata

### R1. Transaction Is Not Retried after BINIT#

**Problem:** If the first transaction of a locked sequence receives a HITM# and DEFER# during the snoop phase it should be retried and the locked sequence restarted. However, if BINIT# is also asserted during this transaction, it will not be retried.

**Implication:** When this erratum occurs, locked transactions will unexpectedly not be retried.

**Workaround:** None identified.

**Status:** For the steppings affected see the *Summary Table of Changes*.

### R2. Invalid Opcode 0FFFh Requires a ModRM Byte

**Problem:** Some invalid opcodes require a ModRM byte (or other following bytes), while others do not. The invalid opcode 0FFFh did not require a ModRM byte in previous generation Intel architecture processors, but does in the Pentium 4 processor.

**Implication:** The use of an invalid opcode 0FFFh without the ModRM byte may result in a page or limit fault on the Pentium 4 processor.

**Workaround:** Use a ModRM byte with invalid 0FFFh opcode.

**Status:** For the steppings affected, see the *Summary Table of Changes*.

### R3. Processor May Hang Due to Speculative Page Walks to Non-Existent System Memory

**Problem:** A load operation that misses the Data Translation Lookaside Buffer (DTLB) will result in a page-walk. If the page-walk loads the Page Directory Entry (PDE) from cacheable memory and that PDE load returns data that points to a valid Page Table Entry (PTE) in uncacheable memory the processor will access the address referenced by the PTE. If the address referenced does not exist the processor will hang with no response from system memory.

**Implication:** Processor may hang due to speculative page walks to non-existent system memory.

**Workaround:** Page directories and page tables in UC memory space which are marked valid must point to physical addresses that will return a data response to the processor.

**Status:** For the steppings affected, see the *Summary Table of Changes*.

## R4. Memory Type of the Load Lock Different from Its Corresponding Store Unlock

**Problem:** A use-once protocol is employed to ensure that the processor in a multi-agent system may access data that is loaded into its cache on a Read-for-Ownership operation at least once before it is snooped out by another agent. This protocol is necessary to avoid a multi-agent livelock scenario in which the processor cannot gain ownership of a line and modify it before that data is snooped out by another agent. In the case of this erratum, split load lock instructions incorrectly trigger the use-once protocol. A load lock operation accesses data that splits across a page boundary with both pages of WB memory type. The use-once protocol activates and the memory type for the split halves get forced to UC. Since use-once does not apply to stores, the store unlock instructions go out as WB memory type. The full sequence on the bus is: locked partial read (UC), partial read (UC), partial write (WB), locked partial write (WB). The use-once protocol should not be applied to load locks.

**Implication:** When this erratum occurs, the memory type of the load lock will be different than the memory type of the store unlock operation. This behavior (load locks and store unlocks having different memory types) does not introduce any functional failures such as system hangs or memory corruption.

**Workaround:** None identified.

**Status:** For the steppings affected, see the *Summary Table of Changes.*

## R5. Machine Check Architecture Error Reporting and Recovery May Not Work As Expected

**Problem:** When the processor detects errors it should attempt to report and/or recover from the error. In the situations described below, the processor does not report and/or recover from the error(s) as intended.

- When a transaction is deferred during the snoop phase and subsequently receives a Hard Failure response, the transaction should be removed from the bus queue so that the processor may proceed. Instead, the transaction is not properly removed from the bus queue, the bus queue is blocked, and the processor will hang.
- When a hardware prefetch results in an uncorrectable tag error in the L2 cache, MC0_STATUS.UNCOR and MC0_STATUS.PCC are set but no Machine Check Exception (MCE) is signaled. No data loss or corruption occurs because the data being prefetched has not been used. If the data location with the uncorrectable tag error is subsequently accessed, an MCE will occur. However, upon this MCE, or any other subsequent MCE, .the information for that error will not be logged because MC0_STATUS.UNCOR has already been set and the MCA status registers will not contain information about the error which caused the MCE assertion but instead will contain information about the prefetch error event.
- When the reporting of errors is disabled for Machine Check Architecture (MCA) Bank 2 by setting all MC2_CTL register bits to 0, uncorrectable errors should be logged in the IA32_MC2_STATUS register but no machine-check exception should be generated. Uncorrectable loads on bank 2, which would normally be logged in the IA32_MC2_STATUS register, are not logged.
- When one-half of a 64-byte instruction fetch from the L2 cache has an uncorrectable error and the other 32-byte half of the same fetch from the L2 cache has a correctable error, the

processor will attempt to correct the correctable error but cannot proceed due to the uncorrectable error. When this occurs the processor will hang.

- When an L1 cache parity error occurs, the cache controller logic should write the physical address of the data memory location that produced that error into the IA32_MC1_ADDR REGISTER (MC1_ADDR). In some instances of a parity error on a load operation that hits the L1 cache, the cache controller logic may write the physical address from a subsequent load or store operation into the IA32_MC1_ADDR register.
- When an error exists in the tag field of a cache line such that a request for ownership (RFO) issued by the processor hits multiple tag fields in the L2 cache (the correct tag and the tag with the error) and the accessed data also has a correctable error, the processor will correctly log the multiple tag match error but will hang when attempting to execute the machine check exception handler.
- If a memory access receives a machine check error on both 64 byte halves of a 128-byte L2 cache sector, the IA32_MC0_STATUS register records this event as multiple errors, i.e., the valid error bit and the overflow error bit are both set indicating that a machine check error occurred while the results of a previous error were in the error-reporting bank. The IA32_MC1_STATUS register should also record this event as multiple errors but instead records this event as only one correctable error.
- The overflow bit should be set to indicate when more than one error has occurred. The overflow bit being set indicates that more than one error has occurred. Because of this erratum, if any further errors occur, the MCA overflow bit will not be updated, thereby incorrectly indicating only one error has been received.
- If an I/O instruction (IN, INS, REP INS, OUT, OUTS, or REP OUTS) is being executed, and if the data for this instruction becomes corrupted, the processor will signal a Machine Check Exception (MCE). If the instruction is directed at a device that is powered down, the processor may also receive an assertion of SMI#. Since MCEs have higher priority, the processor will call the MCE handler, and the SMI# assertion will remain pending. However, while attempting to execute the first instruction of the MCE handler, the SMI# will be recognized and the processor will attempt to execute the SMM handler. If the SMM handler is successfully completed, it will attempt to restart the I/O instruction, but will not have the correct machine state due to the call to the MCE handler. This can lead to failure of the restart and shutdown of the processor.
- If PWRGOOD is de-asserted during a RESET# assertion causing internal glitches, the MCA registers may latch invalid information.
- If RESET# is asserted, then de-asserted, and reasserted, before the processor has cleared the MCA registers, then the information in the MCA registers may not be reliable, regardless of the state or state transitions of PWRGOOD.
- If MCERR# is asserted by one processor and observed by another processor, the observing processor does not log the assertion of MCERR#. The Machine Check Exception (MCE) handler called upon assertion of MCERR# will not have any way to determine the cause of the MCE.
- The Overflow Error bit (bit 62) in the IA32_MC0_STATUS register indicates, when set, that a machine check error occurred while the results of a previous error were still in the error reporting bank (i.e. The Valid bit was set when the new error occurred). If an uncorrectable error is logged in the error-reporting bank and another error occurs, the overflow bit will not be set.

- The MCA Error Code field of the IA32_MC0_STATUS register gets written by a different mechanism than the rest of the register. For uncorrectable errors, the other fields in the IA32_MC0_STATUS register are only updated by the first error. Any further errors that are detected will update the MCA Error Code field without updating the rest of the register, thereby leaving the IA32_MC0_STATUS register with stale information.
- When a speculative load operation hits the L2 cache and receives a correctable error, the IA32_MC1_Status Register may be updated with incorrect information. The IA32_MC1_Status Register should not be updated for speculative loads.
- The processor should only log the address for L1 parity errors in the IA32_MC1_Status register if a valid address is available. If a valid address is not available, the Address Valid bit in the IA32_MC1_Status register should not be set. In instances where an L1 parity error occurs and the address is not available because the linear to physical address translation is not complete or an internal resource conflict has occurred, the Address Valid bit is incorrectly set.
- The processor may hang when an instruction code fetch receives a hard failure response from the system bus. This occurs because the bus control logic does not return data to the core, leaving the processor empty. IA32_MC0_STATUS MSR does indicate that a hard fail response occurred.
- The processor may hang when the following events occur and the machine check exception is enabled, CR4.MCE=1. A processor that has it's STPCLK# pin asserted will internally enter the Stop Grant State and finally issue a Stop Grant Acknowledge special cycle to the bus. If an uncorrectable error is generated during the Stop Grant process it is possible for the Stop Grant special cycle to be issued to the bus before the processor vectors to the machine check handler. Once the chipset receives its last Stop Grant special cycle it is allowed to ignore any bus activity from the processors. As a result, processor accesses to the machine check handler may not be acknowledged, resulting in a processor hang.

**Implication:** The processor is unable to correctly report and/or recover from certain errors.

**Workaround:** None identified.

**Status:** For the steppings affected, see the *Summary Table of Changes*.

## R6. Debug Mechanisms May Not Function As Expected

**Problem:** Certain debug mechanisms may not function as expected on the processor. The cases are as follows:

- When the following conditions occur: 1) An FLD instruction signals a stack overflow or underflow, 2) the FLD instruction splits a page-boundary or a 64 byte cache line boundary, 3) the instruction matches a Debug Register on the high page or cache line respectively, and 4) the FLD has a stack fault and a memory fault on a split access, the processor will only signal the stack fault and the debug exception will not be taken.
- When a data breakpoint is set on the ninth and/or tenth byte(s) of a floating point store using the Extended Real data type, and an unmasked floating point exception occurs on the store, the break point will not be captured.
- When any instruction has multiple debug register matches, and any one of those debug registers is enabled in DR7, all of the matches should be reported in DR6 when the processor goes to the debug handler. This is not true during a REP instruction. As an example, during execution of a REP MOVSW instruction the first iteration a load matches DR0 and DR2 and sets DR6 as FFFF0FF5h. On a subsequent iteration of the instruction, a load matches only DR0. The DR6 register is expected to still contain FFFF0FF5h, but the processor will update DR6 to FFFF0FF1h.
- A data breakpoint that is set on a load to uncacheable memory may be ignored due to an internal segment register access conflict. In this case the system will continue to execute instructions, bypassing the intended breakpoint. Avoiding having instructions that access segment descriptor registers, e.g., LGDT, LIDT close to the UC load, and avoiding serialized instructions before the UC load will reduce the occurrence of this erratum.

**Implication:** Certain debug mechanisms do not function as expected on the processor.

**Workaround:** None identified.

**Status:** For the steppings affected, see the *Summary Table of Changes*.

## R7. Cascading of Performance Counters Does Not Work Correctly When Forced Overflow Is Enabled

**Problem:** The performance counters are organized into pairs. When the CASCADE bit of the Counter Configuration Control Register (CCCR) is set, a counter that overflows will continue to count in the other counter of the pair. The FORCE_OVF bit forces the counters to overflow on every non-zero increment. When the FORCE_OVF bit is set, the counter overflow bit will be set but the counter no longer cascades.

**Implication:** The performance counters do not cascade when the FORCE_OVF bit is set.

**Workaround:** None identified.

**Status:** For the steppings affected, see the *Summary Table of Changes*.

## R8. EMON Event Counting of x87 Loads May Not Work As Expected

**Problem:** If a performance counter is set to count x87 loads and floating point exceptions are unmasked, the FPU Operand Data Pointer (FDP) may become corrupted.

**Implication:** When this erratum occurs, the FPU Operand Data Pointer (FDP) may become corrupted.

**Workaround:** This erratum will not occur with floating point exceptions masked. If floating point exceptions are unmasked, then performance counting of x87 loads should be disabled.

**Status:** For the steppings affected, see the *Summary Table of Changes*.

## R9. System Bus Interrupt Messages without Data Which Receive a HardFailure Response May Hang the Processor

**Problem:** When a system bus agent (processor or chipset) issues an interrupt transaction without data onto the system bus and the transaction receives a HardFailure response, a potential processor hang can occur. The processor, which generates an inter-processor interrupt (IPI) that receives the HardFailure response, will still log the MCA error event cause as HardFailure, even if the APIC causes a hang. Other processors, which are true targets of the IPI, will also hang on hardfail-without-data, but will not record an MCA HardFailure event as the cause. If a HardFailure response occurs on a system bus interrupt message with data, the APIC will complete the operation so as not to hang the processor.

**Implication:** The processor may hang.

**Workaround:** None identified.

**Status:** For the steppings affected, see the *Summary Table of Changes.*

## R10. The Processor Signals Page-Fault Exception (#PF) Instead of Alignment Check Exception (#AC) on an Unlocked CMPXCHG8B Instruction

**Problem:** If a Page-Fault Exception (#PF) and Alignment Check Exception (#AC) both occur for an unlocked CMPXCHG8B instruction, then #PF will be flagged.

**Implication:** Software that depends on the Alignment Check Exception (#AC) before the Page-Fault Exception (#PF) will be affected since #PF is signaled in this case.

**Workaround:** Remove the software’s dependency on #AC having precedence over #PF. Alternately, correct the page fault in the page fault handler and then restart the faulting instruction

**Status:** For the stepping affected, see the *Summary Table of Changes.*

## R11. FSW May Not Be Completely Restored after Page Fault on FRSTOR or FLDENV Instructions

**Problem:** If the FPU operating environment or FPU state (operating environment and register stack) being loaded by an FLDENV or FRSTOR instruction wraps around a 64-KB or 4-GB boundary and a page fault (#PF) or segment limit fault (#GP or #SS) occurs on the instruction near the wrap boundary, the upper byte of the FPU status word (FSW) might not be restored. If the fault handler does not restart program execution at the faulting instruction, stale data may exist in the FSW.

**Implication:** When this erratum occurs, stale data will exist in the FSW.

**Workaround:** Ensure that the FPU operating environment and FPU state do not cross 64-KB or 4-GB boundaries. Alternately, ensure that the page fault handler restarts program execution at the faulting instruction after correcting the paging problem.

**Status:** For the steppings affected, see the *Summary Table of Changes*.

## R12. Processor Issues Inconsistent Transaction Size Attributes for Locked Operation

**Problem:** When the processor is in the Page Address Extension (PAE) mode and detects the need to set the Access and/or Dirty bits in the page directory or page table entries, the processor sends an 8 byte load lock onto the system bus. A subsequent 8 byte store unlock is expected, but instead a 4 byte store unlock occurs. Correct data is provided since only the lower bytes change, however external logic monitoring the data transfer may be expecting an 8-byte store unlock.

**Implication:** No known commercially available chipsets are affected by this erratum.

**Workaround:** None identified.

**Status:** For the steppings affected, see the *Summary Table of Changes*.

## R13. When the Processor Is in the System Management Mode (SMM), Debug Registers May Be Fully Writeable

**Problem:** When in System Management Mode (SMM), the processor executes code and stores data in the SMRAM space. When the processor is in this mode and writes are made to DR6 and DR7, the processor should block writes to the reserved bit locations. Due to this erratum, the processor may not block these writes. This may result in invalid data in the reserved bit locations.

**Implication:** Reserved bit locations within DR6 and DR7 may become invalid.

**Workaround:** Software may perform a read/modify/write when writing to DR6 and DR7 to ensure that the values in the reserved bits are maintained.

**Status:** For the steppings affected, see the *Summary Table of Changes*.

## R14. Shutdown and IERR# May Result Due to a Machine Check Exception on a Hyper-Threading Technology[1] Enabled Processor

**Problem:** When a Machine Check Exception (MCE) occurs due to an internal error, both logical processors on a Hyper-Threading Technology enabled processor normally vector to the MCE handler. However, if one of the logical processors is in the “Wait-for-SIPI” state, that logical processor will not have an MCE handler and will shut down and assert IERR#.

**Implication:** A processor with a logical processor in the “Wait-for-SIPI” state will shut down when an MCE occurs on the other thread.

**Workaround:** None identified.

**Status:** For the steppings affected, see the *Summary Table of Changes.*

## R15. Processor May Hang under Certain Frequencies and 12.5% STPCLK# Duty Cycle

**Problem:** If a system de-asserts STPCLK# at a 12.5% duty cycle, the processor is running below 2 GHz, and the processor thermal control circuit (TCC) on-demand clock modulation is active, the processor may hang. This erratum does not occur under the automatic mode of the TCC.

**Implication:** When this erratum occurs, the processor will hang.

**Workaround:** If use of the on-demand mode of the processor's TCC is desired in conjunction with STPCLK# modulation, then assure that STPCLK# is not asserted at a 12.5% duty cycle.

**Status:** For the steppings affected, see the *Summary Table of Changes*.

## R16. System May Hang if a Fatal Cache Error Causes Bus Write Line (BWL) Transaction to Occur to the Same Cache Line Address as an Outstanding Bus Read Line (BRL) or Bus Read-Invalidate Line (BRIL}

**Problem:** A processor internal cache fatal data ECC error may cause the processor to issue a BWL transaction to the same cache line address as an outstanding BRL or BRIL. As it is not typical behavior for a single processor to have a BWL and a BRL/BRIL concurrently outstanding to the same address, this may represent an unexpected scenario to system logic within the chipset.

**Implication:** The processor may not be able to fully execute the machine check handler in response to the fatal cache error if system logic does not ensure forward progress on the System Bus under this scenario.

**Workaround:** System logic should ensure completion of the outstanding transactions. Note that during recovery from a fatal data ECC error, memory image coherency of the BWL with respect to BRL/BRIL transactions is not important. Forward progress is the primary requirement.

**Status:** For the steppings affected, see the Summary Table of Changes.

## R17. A Write to APIC Registers Sometimes May Appear to Have Not Occurred

**Problem:** In respect to the retirement of instructions, stores to the uncacheable memory-based APIC register space are handled in a non-synchronized way. For example if an instruction that masks the interrupt flag, e.g. CLI, is executed soon after an uncacheable write to the Task Priority Register (TPR) that lowers the APIC priority, the interrupt masking operation may take effect before the actual priority has been lowered. This may cause interrupts whose priority is lower than the initial TPR, but higher than the final TPR, to not be serviced until the interrupt flag is finally cleared, i.e. by STI instruction. Interrupts will remain pending and are not lost.

**Implication:** In this example the processor may allow interrupts to be accepted but may delay their service.

**Workaround:** This non-synchronization can be avoided by issuing an APIC register read after the APIC register write. This will force the store to the APIC register before any subsequent instructions are executed. No commercial operating system is known to be impacted by this erratum.

**Status:** For the steppings affected, see the *Summary Table of Changes*.

## R18. Some Front Side Bus I/O Specifications Are Not Met

**Problem:** The following front side bus I/O specifications are not met:

- The $V_{IH(min)}$ for the GTL+ signals is specified as GTLREF + (0.10 * $V_{CC}$) [V].
- The $V_{IH(min)}$ for the Asynchronous GTL+ signals is specified as Vcc/2 + (0.10 * $V_{CC}$) [V].

**Implication:** This erratum can cause functional failures depending upon system bus activity. It can manifest itself as data parity, address parity, and/or machine check errors.

**Workaround:** Due to this erratum, the system should meet the following voltage levels and processor timings:

- The $V_{IH(min)}$ for GTL+ signals is now GTLREF + (0.20 * $V_{CC}$) [V].
- The $V_{IH(min)}$ for the Asynchronous GTL+ signals is now Vcc/2 + (0.20 * $V_{CC}$) [V].

**Status:** For the steppings affected, see the *Summary Table of Changes*.

## R19. Parity Error in the L1 Cache May Cause the Processor to Hang

**Problem:** If a locked operation accesses a line in the L1 cache that has a parity error, it is possible that the processor may hang while trying to evict the line.

**Implication:** If this erratum occurs, it may result in a system hang. Intel has not observed this erratum with any commercially available software.

**Workaround:** None identified.

**Status:** For the steppings affected, see the *Summary Table of Changes*.

### R20. BPM4# Signal Not Being Asserted According to Specification

**Problem:** BPM4# signal is not being asserted according to the specification. This may cause incorrect operation of In-Target Debuggers, particularly at higher FSB frequencies.

**Implication:** In-Target Debuggers may not function at higher than 133/533 MHz FSB.

**Workaround:** One method is to reduce the FSB common clock frequency to 133 MHz or lower. For higher FSB speeds, In-Target Debuggers have a built-in function (test2010) that tells the hardware to ignore BPM4# assertions. This may degrade the debugger performance but will give correct results.

**Status:** For the steppings affected, see the *Summary Table of Changes*.

### R21. Sequence of Locked Operations Can Cause Two Threads to Receive Stale Data and Cause Application Hang

**Problem:** While going through a sequence of locked operations, it is possible for the two threads to receive stale data. This is a violation of expected memory ordering rules and causes the application to hang.

**Implication:** When this erratum occurs in an HT Technology enabled system, an application may hang.

**Workaround:** It is possible for the BIOS to contain a workaround for this erratum.

**Status:** For the steppings affected, see the Summary Table of Changes.

### R22. A 16-bit Address Wrap Resulting from a Near Branch (Jump or Call) May Cause an Incorrect Address to Be Reported to the #GP Exception Handler

**Problem:** If a 16-bit application executes a branch instruction that causes an address wrap to a target address outside of the code segment, the address of the branch instruction should be provided to the general protection exception handler. It is possible that, as a result of this erratum, that the general protection handler may be called with the address of the branch target.

**Implication:** The 16-bit software environment which is affected by this erratum, will see that the address reported by the exception handler points to the target of the branch, rather than the address of the branch instruction.

**Workaround:** None identified.

**Status:** For the steppings affected, see the *Summary Table of Changes.*

## R23. Bus Locks and SMC Detection May Cause the Processor to Temporarily Hang

**Problem:** The processor may temporarily hang in an HT Technology enabled system if one logical processor executes a synchronization loop that includes one or more locks and is waiting for release by the other logical processor. If the releasing logical processor is executing instructions that are within the detection range of the self -modifying code (SMC) logic, then the processor may be locked in the synchronization loop until the arrival of an interrupt or other event.

**Implication:** If this erratum occurs in an HT Technology enabled system, the application may temporarily stop making forward progress. Intel has not observed this erratum with any commercially available software.

**Workaround:** None identified.

**Status:** For the steppings affected, see the *Summary Table of Changes*.

## R24. PWRGOOD and TAP Signals Maximum Input Hysteresis Higher Than Specified

**Problem:** The maximum input hysteresis for the PWRGOOD and TAP input signals is specified at 350 mV. The actual value could be as high as 800 mV.

**Implication:** The PWRGOOD and TAP inputs may switch at different levels than previously documented specifications. Intel has not observed any issues in validation or simulation as a result of this erratum.

**Workaround:** None identified.

**Status:** For the steppings affected, see the *Summary Table of Changes.*

## R25. Incorrect Physical Address Size Returned by CPUID Instruction

**Problem:** The CPUID instruction Function 80000008H (Extended Address Sizes Function) returns the address sizes supported by the processor in the EAX register. This Function returns an incorrect physical address size value of 40 bits. The correct physical address size is 36 bits.

**Implication:** Function 80000008H returns an incorrect physical address size value of 40 bits.

**Workaround:** None identified.

**Status:** For the steppings affected, see the Summary *Table of Changes.*

## R26. Incorrect Debug Exception (#DB) May Occur When a Data Breakpoint Is Set on an FP Instruction

**Problem:** The default Microcode Floating Point Event Handler routine executes a series of loads to obtain data about the FP instruction that is causing the FP event. If a data breakpoint is set on the instruction causing the FP event, the load in the microcode routine will trigger the data breakpoint resulting in a Debug Exception.

**Implication:** An incorrect Debug Exception (#DB) may occur if data breakpoint is placed on an FP instruction. Intel has not observed this erratum with any commercially available software or system.

**Workaround:** None identified.

**Status:** For the steppings affected, see the Summary *Table of Changes.*

## R27. xAPIC May Not Report Some Illegal Vector Errors

**Problem:** The local xAPIC has an Error Status Register, which records all errors. The bit 6 (the Receive Illegal Vector bit) of this register, is set when the local xAPIC detects an illegal vector in a received message. When an illegal vector error is received on the same internal clock that the error status register is being written (due to a previous error), bit 6 does not get set and illegal vector errors are not flagged

**Implication:** The xAPIC may not report some Illegal Vector errors when they occur at approximately the same time as other xAPIC errors. The other xAPIC errors will continue to be reported.

**Workaround:** None identified

**Status:** For the stepping affected, see the *Summary Table of Changes.*

## R28. Enabling No-Eviction Mode (NEM) May Prevent the Operation of the Second Logical Processor in a Hyper-Threading Technology Enabled Processor

**Problem:** In an HT Technology enabled system, when NEM is enabled by setting bit 0 of MSR 080h (IA32_BIOS_CACHE_AS_RAM), the second logical processor may fail to wake up from "Wait-for-SIPI" state.

**Implication:** In an HT Technology enabled system, the second logical processor may not respond to SIPI. The OS will continue to operate but with fewer logical processors than expected.

**Workaround:** It is possible for the BIOS to contain a workaround for this erratum.

**Status:** For the stepping affected, see the *Summary Table of Changes.*

## R29. Incorrect Duty Cycle is Chosen when On-Demand Clock Modulation is Enabled in a Processor Supporting Hyper-Threading Technology

**Problem:** When a processor supporting Hyper-Threading Technology enables On-Demand Clock Modulation on both logical processors, the processor is expected to select the lowest duty cycle of the two potentially different values. When one logical processor enters the AUTOHALT state, the duty cycle implemented should be unaffected by the halted logical processor. Due to this erratum, the duty cycle is incorrectly chosen to be the higher duty cycle of both logical processors.

**Implication:** Due to this erratum, higher duty cycle may be chosen when the On-Demand Clock Modulation is enabled on both logical processors.

**Workaround:** None identified at this time

**Status:** For the stepping affected, see the *Summary Table of Changes.*

## R30. Memory Aliasing of Pages as Uncacheable Memory Type and Write Back (WB) May Hang the System

**Problem:** When a page is being accessed as either Uncacheable (UC) or Write Combining (WC) and WB, under certain bus and memory timing conditions, the system may loop in a continual sequence of UC fetch, implicit writeback, and Request For Ownership (RFO) retries.

**Implication:** This erratum has not been observed in any commercially available operating system or application. The aliasing of memory regions, a condition necessary for this erratum to occur, is documented as being unsupported in the *IA-32 Intel® Architecture Software Developer's Manual*, Volume 3, section 10.12.4, Programming the PAT. However, if this erratum occurs the system may hang.

**Workaround:** The pages should not be mapped as either UC or WC and WB at the same time.

**Status:** For the stepping affected, see the *Summary Table of Changes.*

## R31. Interactions Between the Instruction Translation Lookaside Buffer (ITLB) and the Instruction Streaming Buffer May Cause Unpredictable Software Behavior

**Problem:** Complex interactions within the instruction fetch/decode unit may make it possible for the processor to execute instructions from an internal streaming buffer containing stale or incorrect information.

**Implication:** When this erratum occurs, an incorrect instruction stream may be executed resulting in unpredictable software behavior.

**Workaround:** It is possible for the BIOS to contain a workaround for this erratum.

**Status:** For the stepping affected, see the *Summary Table of Changes.*

§

# Specification Changes

The Specification Changes listed in this section apply to the following documents:

- *Intel® Pentium® 4 Processor on 90 nm Process Datasheet*

All Specification Changes will be incorporated into a future version of the appropriate Pentium 4 processor documentation.

There are no specification changes in this Specification Update revision.

§

# Specification Clarifications

The Specification Clarifications listed in this section apply to the following documents:

- *Intel® Pentium® 4 Processor on 90 nm Process* Datasheet

All Specification Clarifications will be incorporated into a future version of the appropriate Pentium 4 processor documentation.

There are no specification clarifications in this Specification Update revision.

§

# Documentation Changes

The Documentation Changes listed in this section apply to the following documents:

- *Intel® Pentium® 4 Processor on 90 nm Process Datasheet*

All Documentation Changes will be incorporated into a future version of the appropriate Pentium 4 processor documentation.

**Note: Documentation changes for IA-32 Intel® Architecture Software Developer's Manual volumes 1, 2A, 2B, 3 and Intel® Extended Memory 64 Technology Software Developer's Guide volumes 1, 2 will be posted in a separate document *IA-32 Intel® Architecture and Intel® Extended Memory 64 Technology Software Developer's Manual Documentation Changes*. Follow the link below to become familiar with this file.**

http://developer.intel.com/design/pentium4/specupdt/252046.htm

There are no documentation changes in this Specification Update revision.

§